Universidad de San Carlos de Guatemala
Lab. IPC1
Ingeniero: Inge. Veliz
Auxiliar: Lester López

Manual Técnico

Monroy Maldonado
Christopher Alejandro
Sección: D
Fecha: 16/02/2025

Introducción

En este manual se describe como he implementado mis conocimientos y también investigación para la creación de un juego de sopa de letras con muchas funciones, además de usar una estructura muy ordenada para entender mejor el sistema. Todo este proyecto fue realizado con Java en NetBeans.

Requisitos técnicos

Hardware

Procesador: 1 GHz o superior

Memoria RAM: 512 MB mínimo

Espacio en Disco: 50 MB mínimo

Software

Java Development Kit (JDK) 8 o superior

NetBeans, Eclipse o cualquier IDE compatible con Java

Sistema operativo: Windows, macOS o Linux

Descripción Técnica

Para este proyecto utilizamos las librerías únicamente Random y Sccaner. Estas nos permiten leer lo que el usuario escriba y el Random funciona para dar caracteres, letras, números al azar y los usamos para la creación del tablero.

Atributos y declaración:

En este apartado es donde se declara y se almacena cada una de las variables que utilizaremos dentro de nuestro programa. Dentro de los paréntesis se explica que es cada una de las variables declaradas

private String[] palabras; (Vector para almacenar palabras)

private String[] historial; (Vector para almacenar historial de partidas)

private int[] puntuaciones; (Vector para almacenar puntuaciones)

private int puntos; (Contador de puntos)

private int fallos; (Contador de fallos)

private char[][] tablero; (Creador del tablero)

private int numPalabras; (Contador de palabras)

private int numHistorial; (Contador de historial)

private int numPuntuaciones; (Contador para el numero de puntuaciones)

Método Constructor

En este método creamos el tablero que será de 25x25. Además de asiganrle de cuanto será el máximo de capacidad de cada vector, donde se pondrá las puntuaciones, las palabras y el historial.

```
public Practica1() {
        palabras = new String[5]; // Capacidad máxima de 5 palabras
        historial = new String[100]; // Capacidad máxima de 100 partidas
        puntuaciones = new int[100]; // Capacidad máxima de 100 puntuaciones
        tablero = new char[25][25];
        for (int i = 0; i < 25; i++) {
            for (int j = 0; j < 25; j++) {
                tablero[i][j] = '-';
            }
        }
        puntos = 25;
        fallos = 0;
        numPalabras = 0;
        numHistorial = 0;
        numPuntuaciones = 0;
    }
```

El método main

En este método es donde inicia la selección del menú y se invoca o se manda a llamar al método constructor, este método es la base del sistema ya que depende la opción del usuario es donde se llama cada uno de los métodos por separado y también donde se sale del juego. En este apartado usamos un ciclo do-while para que el proceso suceda por lo menos una vez y también use un switch and case para la selección del menú y sea más eficiente que usar if por si solos.

```java
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    Practical juego = new Practical();
    int opcion;

    System.out.print("Bienvenidos a mi programa :)");
    System.out.print("Las reglas son simples");
    System.out.print("Agrege las palabras en Gestion de palabras y luego regresar al menu");
    System.out.print("Por ultimo selecciona Empezar nueva partida y Diviertete");
    do {
        System.out.println(" Selecciona una de las opciones ");
        System.out.println("1. Empezar Nueva Partida");
        System.out.println("2. Mostrar Historial de Partidas");
        System.out.println("3. Mostrar Puntuaciones");
        System.out.println("4. Gestionar Palabras");
        System.out.println("5. Mostrar Informacion del Estudiante");
        System.out.println("6. Salir");
        System.out.print("Seleccione una opcion: ");
        opcion = scanner.nextInt();
```

```
        switch (opcion) {
            case 1:
                juego.comenzar();
                break;
            case 2:
                juego.mostrarHistorial();
                break;
            case 3:
                juego.puntuaciones();
                break;
            case 4:
                juego.gestionar();
                break;
            case 5:
                juego.estudiante();
                break;
            case 6:
                System.out.println("Saliendo del juego. Gracias por jugar");
                break;
            default:
                System.out.println("Opción no válida. Intente de nuevo.");
        }
    } while (opcion != 6);
}
```

Método Gestionar

En este método es el principal para la creación, modificación y eliminaciones de las palabras escritas por el usuario. Este también tiene un menú para entrar a los diferentes métodos, además de tener una opción para volver al menú principal. Utilice otro do-while para que tenga que pasar por ese proceso 1 vez hasta que sea diferente de 5 repite todo.

```
public void gestionar() {
    Scanner scanner = new Scanner(System.in);
    int opcion;
    do {
        System.out.println("Gestion de Palabras");
        System.out.println("1. Insertar Palabra");
        System.out.println("2. Modificar Palabra");
        System.out.println("3. Eliminar Palabra");
        System.out.println("4. Mostrar Palabras");
        System.out.println("5. Regresar al Menu Principal");
        System.out.print("Seleccione una opcion: ");
        opcion = scanner.nextInt();

        switch (opcion) {
            case 1:
                insertar();
                break;
            case 2:
                modificar();
                break;
            case 3:
                eliminar();
                break;
            case 4:
                mostrar();
                break;
            case 5:
                System.out.println("Regresando al menú principal...");
                break;
            default:
                System.out.println("Opción no válida. Intente de nuevo.");
                break;
        }
    } while (opcion != 5);
}
```

Método Insertar

En este método el programa lee la palabra que el usuario ingresa además que llama al método validar palabra (más adelante se profundizara en este método). La función ToLowerCase lee la palabra en minúscula o mayúscula. Agrega un valor al contador NumPalabras, y al ingresar la palabra deseada depende si cumple los requisitos lanza un texto que la palabra fue ingresada de manera correcta o incorrecta.

```
public void insertar() {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Ingrese una palabra de 6 a 15 caracteres: ");
    String palabra = scanner.nextLine();
    if (validarPalabra(palabra)) {
        palabras[numPalabras++] = palabra.toLowerCase(); // Agregar pala
        System.out.println("Palabra insertada correctamente.");
    } else {
        System.out.println("La palabra no cumple con los requisitos.");
    }
}
```

Función Boleana publica validarPalabra

En esta función recibe la palabra que fue ingresada por el usuario y detecta si cumple con la condición que tenga 6 a 15 letras y devuelve el valor.

```
public boolean validarPalabra(String palabra) {
    return palabra.length() >= 6 && palabra.length() <= 15;
}
```

Metodo eliminar

Esta función permite eliminar la palabra que el usuario quiera escribiéndola el mismo así esta seguro de cual palabra necesita borrar. Manda a llamar al método eliminarPalabra (Se definirá que es más adelante). Además utilizamos otro toLowerCase para leer la palabra en minúsculas.

```
public void eliminar() {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Ingrese la palabra que desea eliminar: ");
    String palabra = scanner.nextLine().toLowerCase(); // Convertir a m
    if (eliminarPalabra(palabra)) {
        System.out.println("Palabra eliminada correctamente.");
    } else {
        System.out.println("La palabra no se encuentra en la lista.");
    }
}
```

Función Boleana eliminarPalabra

En esta función recibe los strings palabra que el usuario a ingresado. Usa un ciclo for para contar las palabras y recibir los datos, elimina la palabra. El otro ciclo for es para correr una posición a la izquierda al vector de las palabra. Si es verdadero regresa el vector si la palabra sino significa que no la encontro y quita 1 al contador del numPalabras.

```
public boolean eliminarPalabra(String palabra) {
    for (int i = 0; i < numPalabras; i++) {
        if (palabras[i].equals(palabra)) {
            // Mover las palabras hacia la izquierda
            for (int j = i; j < numPalabras - 1; j++) {
                palabras[j] = palabras[j + 1];
            }
            palabras[--numPalabras] = null; // Reducir el tamaño
            return true;
        }
    }
    return false;
}
```

Método mostar

En este método muestra las palabras ingresadas por el usuario usa un ciclo for para tomar las posiciones del vector y agregar la palabra con un mensaje al usuario en la consola.

```
public void mostrar() {
    System.out.println("Palabras ingresadas por el usuario:");
    for (int i = 0; i < numPalabras; i++) {
        System.out.println("  " + palabras[i]);
    }
}
```

Metodo modificar

En este método le pregunta al usuario que palabra es la que desea modificar. EL ciclo for recorre el vector de palabras para encontrar la palabra ingresda previamente por el usuario. Después le pregunta cual es la palabra por la que desea modificarla esta también debe tener de 6 a 15 letras. Usa la función validarPalabra para saber si cumple los requisitos y de igual forma también lo lee en minúsculas. Tiene un sistema de error por si la palabra no cumple con los requisitos y da un mensaje que se ha cambiado si cumple

```java
public void modificar() {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Ingrese la palabra a modificar: ");
    String palabraA = scanner.nextLine().toLowerCase(); // Convertir a minúsculas
    for (int i = 0; i < numPalabras; i++) {
        if (palabras[i].equals(palabraA)) {
            System.out.print("Ingrese la nueva palabra (debe tener de 6 a 15 caracteres): ");
            String nuevaPalabra = scanner.nextLine();
            if (validarPalabra(nuevaPalabra)) {
                palabras[i] = nuevaPalabra.toLowerCase(); // Modificar la palabra en minúsculas
                System.out.println("Palabra modificada correctamente.");
            } else {
                System.out.println("La nueva palabra no cumple con los requisitos.");
            }
            return;
        }
    }
    System.out.println("La palabra a modificar no se encuentra en la lista.");
}
```

Metodo de colocarPalabra

En este método es donde se colocan las palabras aleatorias para completar el tablero de 25x25, además de mostrar las palabras que el usuario ingreso estas pueden ser en vertical u horizontal. Use un while para hacer un bucle y que de las letras aleatorias en filas y columnas. EL siguiente if es solo para buscar la palabra que si esta en la posición horizontal o en la vertical depende de donde este colocada la palabra de manera aleatoria.

```java
public void colocarPalabras() {
    Random random = new Random();
    for (int i = 0; i < numPalabras; i++) {
        String palabra = palabras[i].toLowerCase(); // Convertir a minúsculas
        boolean colocada = false;
        while (!colocada) {
            int fila = random.nextInt(25);
            int columna = random.nextInt(25);
            boolean horizontal = random.nextBoolean();
            if (horizontal && columna + palabra.length() <= 25) {
                // Colocar horizontalmente
                boolean col = true;
                for (int j = 0; j < palabra.length(); j++) {
                    if (tablero[fila][columna + j] != '-') {
                        col = false;
                        break;
                    }
                }
                if (col) {
                    for (int j = 0; j < palabra.length(); j++) {
                        tablero[fila][columna + j] = palabra.charAt(j);
                    }
                    colocada = true;
                }
            } else if (!horizontal && fila + palabra.length() <= 25) {
                // Colocar verticalmente
                boolean puedeColocar = true;
                for (int j = 0; j < palabra.length(); j++) {
                    if (tablero[fila + j][columna] != '-') {
                        puedeColocar = false;
                        break;
                    }
                }
                if (puedeColocar) {
                    for (int j = 0; j < palabra.length(); j++) {
                        tablero[fila + j][columna] = palabra.charAt(j);
                    }
                    colocada = true;
                }
            }
        }
    }
    // Rellenar el resto del tablero con letras aleatorias
    for (int i = 0; i < 25; i++) {
        for (int j = 0; j < 25; j++) {
            if (tablero[i][j] == '-') {
                tablero[i][j] = (char) ('A' + random.nextInt(26));
            }
        }
    }
}
```

Método marcarPalabraEncontrada

Convierte la palabra en minúscula y con los ciclos recorre la matriz además de que verifica si esta en vertical u horizontal la palabra y recorre las letras de la sopa de letras para ver si están las que ingreso el usuario al momento de jugar, si se encuentra la cambia por # de primero lo recorre de manera horizontal sino cambia la búsqueda y lo busca de manera vertical. Y usa el return para no marcar la palabra más veces

```
public void marcarPalabraEncontrada(String palabra) {
    palabra = palabra.toLowerCase(); // Convertir a minúsculas
    for (int i = 0; i < 25; i++) {
        for (int j = 0; j < 25; j++) {
            if (tablero[i][j] == palabra.charAt(0)) {
                // Verificar si la palabra está horizontal
                if (j + palabra.length() <= 25) {
                    boolean encontrada = true;
                    for (int k = 0; k < palabra.length(); k++) {
                        if (tablero[i][j + k] != palabra.charAt(k)) {
                            encontrada = false;
                            break;
                        }
                    }
                    if (encontrada) {
                        for (int k = 0; k < palabra.length(); k++) {
                            tablero[i][j + k] = '#';
                        }
                        return;
                    }
                }
                // Verificar si la palabra está vertical
                if (i + palabra.length() <= 25) {
                    boolean encontrada = true;
                    for (int k = 0; k < palabra.length(); k++) {
                        if (tablero[i + k][j] != palabra.charAt(k)) {
                            encontrada = false;
                            break;
                        }
                    }
                    if (encontrada) {
                        for (int k = 0; k < palabra.length(); k++) {
                            tablero[i + k][j] = '#';
                        }
```

```
                        return;
                    }
                }
            }
        }
    }
```

Método mostrarTablero

Muestra un texto para decir que es el tablero de juego. Usa ciclos for para recorrer el tablero y muestra las letras y palabras

```
public void mostrarTablero() {
    System.out.println("Tablero de juego");
    for (int i = 0; i < 25; i++) {
        for (int j = 0; j < 25; j++) {
            System.out.print(tablero[i][j] + " ");
        }
        System.out.println();
    }
}
```

Metodo comenzar

Lo primero que hace es un reconocimiento de palabras y si no hay palabras lanza un mensaje de error, luego llama al método colocarPalabras para colocar las palabras en posiciones aleatorias. Después verifica con un ciclo while si la persona no ha cometido 4 errores para que el juego continúe, además llama al método mostrarTablero para mostrar en pantalla el tablero con las palabras del usuario en posiciones aleatorias y las palabra al azar para rellenar. SI el usuario encuentra la palabra llama al método marcarPalabraEncontrada y suma los puntos que depende la cantidad de letras que tenga la palabra ingresada por el usuario. Sino está en el tablero agrega 1 error y resta los puntos. Lanza diferentes mensaje si se encontraron las palabras o si fallaron 4 veces y al final muestra el puntaje que hizo el jugador. Y guarda la puntuación[] y el historial[].

```
public void comenzar() {
    if (numPalabras == 0) {
        System.out.println("No hay palabras para jugar. Agregue palabras primero.");
        return;
    }
    colocarPalabras();
    Scanner scanner = new Scanner(System.in);
    while (fallos < 4 && numPalabras > 0) {
        mostrarTablero();
        System.out.print("Ingrese una palabra: ");
        String palabraIngresada = scanner.nextLine().toLowerCase(); // Convertir a minúsculas
        if (buscarPalabra(palabraIngresada)) {
            marcarPalabraEncontrada(palabraIngresada);
            puntos += palabraIngresada.length();
            eliminarPalabra(palabraIngresada);
            System.out.println("¡Palabra encontrada! Puntos: " + puntos);
        } else {
            fallos++;
            puntos -= 5;
            System.out.println("Palabra no encontrada. Fallos: " + fallos + ", Puntos: " + puntos);
        }
    }
    if (fallos >= 4) {
        System.out.println("Has alcanzado el límite de fallos. Fin del juego.");
    } else {
        System.out.println("¡Has encontrado todas las palabras! Puntos finales: " + puntos);
    }
    historial[numHistorial++] = " - Puntos: " + puntos;
    puntuaciones[numPuntuaciones++] = puntos;
}
```

Función buscarPalabra

El LowerCase hace que ni importa si la palabra es en mayúsculas o minúsculas y compara las palabras para que sea igual a otras y devuelve el factor true sino devuelve false.

```
public boolean buscarPalabra(String palabra) {
    palabra = palabra.toLowerCase();
    for (int i = 0; i < numPalabras; i++) {
        if (palabras[i].equals(palabra)) {
            return true;
        }
    }
    return false;
}
```

Método mostrarHistorial

Muestra un texto que dice que es el historial de partidas, el for cuenta las veces que se ha jugado y en qué posición para mostrar la puntuación y en que partida fue.

```
public void mostrarHistorial() {
    System.out.println("Historial de partidas:");
    for (int i = 0; i < numHistorial; i++) {
        System.out.println(historial[i]);
    }
}
```

Método estudiante

Muestra el texto del nombre del estudiante, su carnet y su sección. Y regresa al menú principal

```
public void estudiante() {
    System.out.println("Nombre: Christopher Alejandro Monroy Maldonado");
    System.out.println("Carnet: 202400860");
    System.out.println("Seccion: D");
}
```

Método Puntuaciones

Muestra las puntuaciones que se ha conseguido en cada juego. Con el contador y los muestra en la pantalla. Usa fors anidados para buscar cual fue la puntuación más alta para mostrarla en pantalla y luego busca la 2da más grande y así en sucesión hasta llegar a la última.

```
public void puntuaciones() {
    System.out.println("Puntuaciones más altas:");
for (int i = 0; i < numPuntuaciones - 1; i++) {
    for (int j = 0; j < numPuntuaciones - 1 - i; j++) {
        if (puntuaciones[j] < puntuaciones[j + 1]) {
            int temp = puntuaciones[j];
            puntuaciones[j] = puntuaciones[j + 1];
            puntuaciones[j + 1] = temp;
        }
    }
}

for (int i = 0; i < numPuntuaciones; i++) {
    System.out.println((i + 1) + ". " + puntuaciones[i]);
}
}
```

Método jugarV

En este método lo que se hace es pedirle al jugador su nombre y reinicia el tablero para poder volver a jugar.

```
public void jugarV() {
    System.out.println("Antes de Iniciar Dame tu nombre");
    Scanner scanner = new Scanner(System.in);
    String Nombre= scanner.nextLine();
        comenzar(); // Iniciar una nueva partida
        puntos = 25;
        fallos = 0;
        for (int i = 0; i < 25; i++) {
            for (int j = 0; j < 25; j++) {
                tablero[i][j] = '-';
            }
        }
}
```

Conclusión

Esta práctica fue muy importante para aprender más acerca de los ciclos, además de los arreglos y vectores. Fue muy productivo hacer un sistema complejo que use muchos ciclos, bucles, procesos además de tener un nivel alto de comprensión de programación y lógica. Esta práctica fue de mucha ayuda para familiarizarse con el lenguaje de Java